Transceiver Packet Format

The packets used in the transceiver modules are divided into two main classes. The first class is referred as the control packet and it is used by the radio packet controller (RPC). The second class is the data packet, which includes all the user-related data. You will need to send only the data packet part i.e. the number of bytes, the Target ID, Sender ID and the data respectively.

BYTE	TYPE	FUNCTION
1	Byte	Number of bytes in the packet
2	Byte	Destination ID
3	Byte	Sender ID
4-27	Word	Data such as (x, y) coordinate, speed, orientation, time

Figure 1: Content of the data packet

Sender ID, Destination ID: Since the RPC does not generate routing information to preserve versatility, some network-related information such as source and destination addresses is needed in the packet. Each agent in the network has a unique ID, which is used both to address the agent and to indicate the origin of the message. Whenever a new peer is added, it is assigned an ID. Since one byte is reserved for this information, the network is capable of serving to 255 agents. The Initially the network will consist of four mobile robots and four PCs. The following ID's are assigned to each agent as seen in Figure 2.

Agent	ID
Mobile Robot 1 (Mobot 1)	1
Mobot 2	2
Mobot 3	4
Mobot 4	8
PC1	16
PC2	32
PC3	64
PC4	128

Figure 2 The ID's of the agents present in the network

Example 1:

Assume that you are the operator of Mobot1 and PC1. You want to send your position information to Mobot4. Write a code that sends these data.

Hint:

Position means x_right (it is integer type and occupies 2 bytes in the memory of DSP) and x_left.

Solution:

Assume x_right and x_left have been already calculated.

```
/**************Code that is used to send data****************/

SendByte(5)        /* Totally 5 bytes will be sent */
temp = x_right & 0x00FF;   /*temp is integer type variable */
SendByte(temp)    /* The least significant byte of x_right is sent */
temp = (x_right & 0xFF00) >> 8;
SendByte(temp)    /* The most significant byte of x_right is sent */
temp = x_left & 0x00FF;   /*temp is integer type variable */
SendByte(temp)    /* The least significant byte of x_left is sent */
temp = (x_left & 0xFF00) >> 8;
SendByte(temp)    /* The most significant byte of x_left is sent */

/*****************************************************/
```

Example 2:

Assume that, your task is to perform a cooperative task with Mobot4. In order to be successful in this scenario, Mobot4 sends you two-byte data, from which you can find out its orientation, and speed. Being aware of the fact, that you are in a lab where two other groups are performing experiments, write a code that takes the data from Mobot4.

Hint 1: Whenever new data arrives to the transceiver of your mobot, the program will enter an external interrupt service routine. You will put your code in this routine.

Hint 2: A function called ReceiveByte() which takes single byte from the transceiver is available for you.

Solution

```
/****************************************/

count = ReceiveByte();          /* Take the first byte. According to the protocol it shows number of bytes in the packet, so we get total number of bytes received by the transceiver. */
Target_ID = ReceiveByte();  /* Check the target of the packet */
if ( Target_ID == MyID)
{
        Sender_ID = ReceiveByte();    /* Take the sender ID */
        angle = ReceiveByte();
        speed = ReceiveByte();
        count = count - 5;                  /* 5 bytes have been taken */
}
ResetTransceiver();                 /* discard the data if it doesn't belong to the receiver */

/*****************************************/
```